8086、8088 微处理器补充材料

2.1 086 的内部结构

8086 微处理器是 Intel 公司分别在 1978 年开发的。8086 均采用双列直插式封装（DIP），共有 40 个引脚。

2.1.1 8086 微处理器内部结构

微处理器 8086 是由算术逻辑单元 ALU、累加器、专用和通用寄存器、指令寄存器、指令译码器及定时和控制电路等组成。按照功能可以将 8086 的内部分成两个部分——总线接口单元 BIU（Bus Interface Unit）和执行单元 EU。如图 2.1所示。

1. 总线接口单元

总线接口单元 BIU 包括 4 个段寄存器、一个指令寄存器、一个与 EU 通信的内部暂存器、先入先出的指令队列、总线控制逻辑和一个用于计算 20 位实际物理地址的加法器Σ。

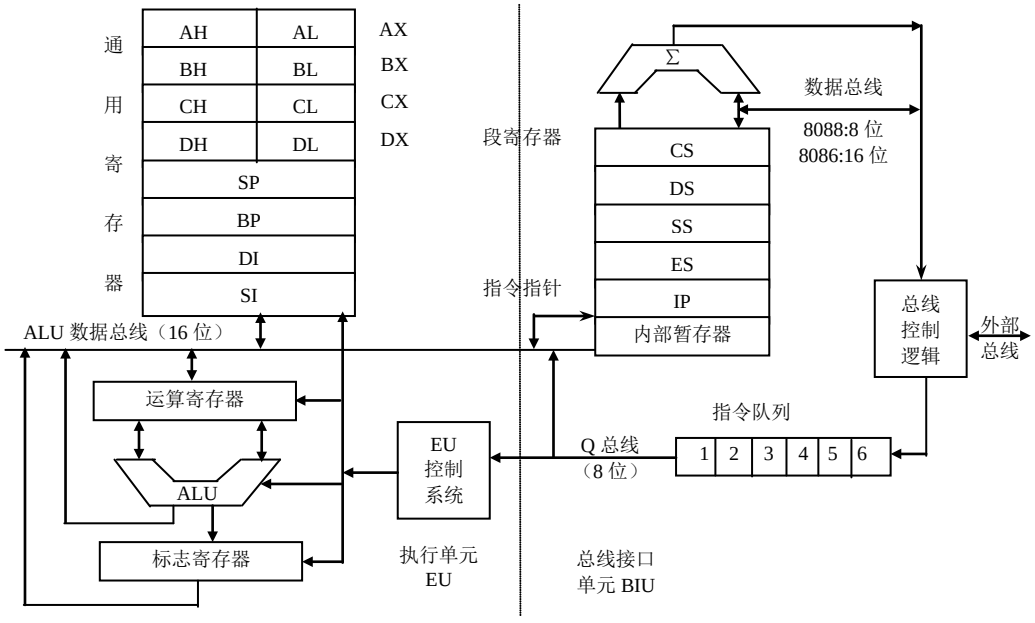

图 2.1 8086 微处理器内部结构示意图

BIU 的主要功能是负责与存储器及 I/O 接口传送信息。一方面，BIU 根据计算得到的地址从内存中取出将要执行的指令到指令队列中；同时，如果指令执行需要数据，BIU 还负责根据计算出的地址，通过总线，到内存或者 I/O 端口中将数据取出，送到 EU 中；另一方面，EU 计算的结果如果需要存储到内存或者输出到 I/O 端口，则也是由 BIU 部件根据计算得到的地址将数据通过总线传到内存或 I/O 端口。

2. 执行单元

执行单元 EU 由 8 个通用寄存器和 1 个标志寄存器、算术逻辑单元 ALU 和 EU 控制系统电路组成。EU 的功能是执行指令。

EU 从指令队列中取出指令代码，将其译码，发出相应的控制信号。从内存或 I/O 端口通过总线和 BIU 来的数据以及从 EU 的通用寄存器中来的数据在 ALU 中进行算术和逻辑运算。运算结果的特征将影响到标志寄存器 FLAGS 的相关位。运算的结果根据指令要求，存入通用寄存器或者通过 BIU 传送到内存或 I/O 端口。另外，EU 也负责计算内存地址，并将该地址传送给 BIU。由此可知，EU 负责所有指令的执行。

3. 总线接口单元与执行单元的配合

BIU 和 EU 虽然是 8086 处理器中的两个部件，但是他们之间是相互配合工作的。EU 会从指令队列中取出指令执行，由于 EU 从指令队列中将指令取走，指令队列会出现空字节，此时，BIU 就从内存中取出后续的指令代码放入队列中；当 EU 需要数据时，BIU 根据 EU 给出的有效地址计算出实际的 20 位物理地址，并从指定的内存单元或外设中取出数据供 EU 使用；运算结束以后，BIU 将运算结果送入指定的内存单元或外设。如果指令队列的所有字节全空，EU 就会等待直到有指令为止。通常，当 8088 的指令队列空出 1 个字节、8086 的指令队列空出 2 个字节的时候，BIU 就会自动执行一次取指令周期，将新指令送入队列。一般情况下，程序是顺序执行，如果遇到跳转指令，BIU 就使指令队列复位，从新地址取出指令，并立即传给 EU 去执行。

BIU 和 EU 这两个功能部件是能够相互独立工作的，再加上 BIU 中有指令队列作为缓冲，因此，在大多数情况下，取指令和执行指令是可以重叠进行的。也就是说，当 EU 在执行第一条指令的同时，BIU 可以去取第二条指令，而当 EU 在执行第二条指令的同时，BIU 可以去取第三条指令，…… 以此类推如图 2.2所示。由此可以看到，大部分时候取指令的时间被执行上一条指令的时间“隐藏”了起来。从宏观上看，BIU 和 EU 两个单元大部分情况下是在并行工作。

图 2.2 取指与执行指令的重叠进行

EU 和 BIU 并行工作，减少了 CPU 为取指令而等待的时间，提高了 CPU 的利用率，加快了整机的运行速度。另外，也降低了对存储器存取速度的要求。

但这种并行也会遇到一个问题，当 BIU 正在取指令的时候，执行上一条指令的 EU 可能由于需要数据或输出运算结果而发出了对总线访问的请求，此时，BIU 要通过总线取指令，EU 也要让 BIU 通过总线取数据或者写数据，这显然造成总线的冲突。因此规定，EU 的总线访问请求必须在 BIU 取指令完毕后才会被得到响应。

2.1.2 8088 和 8086 内部结构上的区别

Intel8088 微处理器的内部结构实质上与 8086 基本相同，其内部也由 BIU 与 EU 两部分组成，两个部分也是并行工作。但两者在 BIU 上还是表现出以下的区别。

- 8086 的指令队列有 6 个字节，而 8088 的指令队列只有 4 个字节。
- 8086 的指令队列空出 2 个字节的时候，BIU 自动执行一次取指令周期。而 8088 是在指令队列空出 1 个字节的时候自动执行一次取指令周期。

- 尽管 8086 和 8088 的内部数据总线都是 16 位，但是与 8086 的 BIU 相连的 CPU 外部的数据总线依然是 16 位，而与 8088 的 BIU 相连的 CPU 外部的数据总线只有 8 位，因此 8088 也称为准 16 位机。

2.2 8086 的引脚信号

在微机系统和微机应用系统的分析与设计时，理解微处理器芯片的引脚功能是很重要的。

本节也是本章的重点和难点之一。

1. 8086 的工作方式

8086 有两种基本的工作方式：最小方式和最大方式。

最小方式是指在系统中只有 8086 一个微处理器，系统中的所有总线控制信号都直接由 8086 产生，因此整个系统中的控制线路最简单。因为本课程实验采用 8086 最小方式，所以本书主要介绍最小方式。

最大方式是相对于最小方式而言的，指系统中含有两个或两个以上的总线主设备，其中一个就是 8086，它为主处理器，其他都是协处理器。在 8086 系统中与其配合的协处理器有数值运算协处理器 8087 和输入/输出协处理器 8089。PC 机采用了最大工作方式。

8086 工作在何种方式，完全由硬件决定。当微处理器引脚的 MN/$\overline{MX}$ 接高电平时，工作在最小方式，接低电平（地）时，则工作在最大方式。

2. 8086 的引脚

8086 外部采用 40 引脚双列直插式封装。图 2.3所示是 8086 引脚图，括号内为最大方式下引脚的定义。

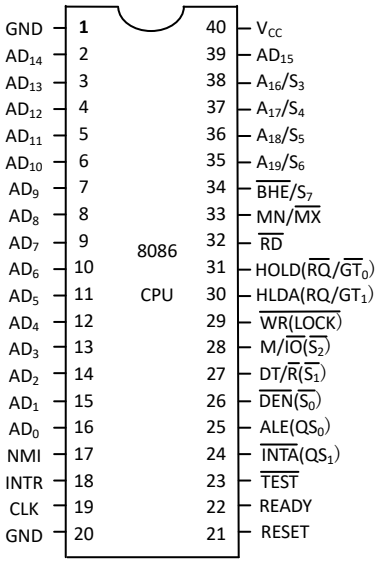

图 2.3 8086 引脚

8086 芯片的各类信号线包括 20 根地址线、16 根（8086）数据线及控制线、状态线、时钟线、电源线和地线等，总数大大超过了 40 根线。因此，为满足封装的要求，部分引脚必须采用一线多用的办法。

注意，在本教材所列信号中，信号名上有短横线或者信号名后面带#后缀，表明该信号

低电平有效。比如$\overline{\text{INTA}}$，IRDY#等。

1） 8086 最大方式和最小方式下公共引脚

- V_{CC} 电源。8088 采用±10%单一+5V 电源。
- AD15~AD0（Address/Data Bus） 地址/数据复用线，双向工作。在总线周期开始的 T1 周期， 作为 20 位地址总线的低 16 位发送地址信息。在总线周期的其他 T 周期，作为数据线读/写 16 位数据。
- A19/S6~A16/S3（Address/Status） 地址/状态复用输出线。S6 恒等于 0；S5 表明中断允许标志位的状态，S5=1 表明 CPU 可以响应可屏蔽中断的请求，S5=0 表明 CPU 禁止一切可屏蔽中断；S4 和 S3 的组合表明当前正在使用的段寄存器，详见
- 表 2.1。
- NMI（Non-Maskable Interrupt） 非屏蔽中断申请输入线。非屏蔽中断申请输入信号必须是一个由低到高的上升沿，这类中断是一种不可用软件屏蔽的中断。
- INTR（Interrupt Request） 可屏蔽中断申请输入线。可屏蔽中断申请信号高电平有效，这类中断可用软件屏蔽。

表 2.1 S4 和 S3 的组合所代表的正在使用的寄存器

S4	S3	当前正在使用的寄存器
0	0	ES
0	1	SS
1	0	CS 或未使用任何段寄存器
1	1	DS

- CLK（Clock） 时钟输入线。该引脚接至时钟发生器 8284 集成电路的输出线，由 8284 提供 8088 所需的 4.77MHz、33%占空比（即 1/3 周期为高电平，2/3 周期为低电平）的系统时钟信号。
- RESET 系统复位信号输入线。RESET 信号高电平有效，8086 要求该信号的有效时间至少为 4 个时钟周期。接通电源或按 RESET 键，都可产生 RESET 信号。CPU 接收到 RESET 信号后，立即停止当前操作，完成内部的复位过程，恢复到机器的起始状态，即 CS=FFFFH，而 IP, DS, ES, SS 以及标志寄存器 FLAGS 被清零，指令队列清空。由于复位后，CS：IP=0FFFFH：0000H，所以 PC 机开机或复位后，都会自动从这个地址开始取指令。
- READY “准备好”信号输入线。准备好信号是由被访问的内存或 I/O 设备发出的响应信号，高电平有效。当其有效时，表示内存或 I/O 设备已准备好，CPU 可以进行数据传送。若内存或 I/O 设备还未准备好，则使 READY 信号为低电平。CPU 采集到低电平的 READY 信号后，自动插入 1 个至多个等待周期 T_w，直到 READY 变为高电平后，CPU 才脱离等待状态，完成数据传送过程。
- $\overline{\text{TEST}}$ 测试信号输入线。当 CPU 执行 WAIT 指令时，每隔 5 个时钟周期对 $\overline{\text{TEST}}$ 引脚进行一次测试。若为高电平，CPU 就仍处于空转状态进行等待，直到 $\overline{\text{TEST}}$ 引脚变为低电平，CPU 结束等待状态，执行下一条指令，以使 CPU 与外部硬件同步。
- $\overline{\text{RD}}$（Read） 读信号输出线。读信号是一个低电平有效的输出信号，当 $\overline{\text{RD}}$ 为低电平时，表明 CPU 正在对内存或外设进行读操作。
- MN/$\overline{\text{MX}}$（Minimum/Maximum Mode Control） 最小/最大方式控制信号输入线。该引脚接至高电平，8086 工作在最小方式；该引脚接至低电平，8086 工作在最大方式。
- $\overline{\text{BHE}}$/S7（Bus High Enable/S7） 总线高允许/S7 状态输出。是高 8 位数据总线允许

信号。这是因为 8086 有 16 条数据线，它可以传送一个字，也可以用高 8 位数据线或低 8 位数据线传送一个字节。 在总线周期开始的 T1 周期， 作为总线高 8 位允许信号，低电平有效。在总线周期的其他 T 周期，该引脚输出状态信号 S7。实际上，S7 时预留的，未被定义。 $\overline{\text{BHE}}$/S7 与 AD0 的不同组合对应了不同的操作，见表 2.2。

表 2.2 $\overline{\text{BHE}}$/S7 与 AD0 的不同组合对应的操作

$\overline{\text{BHE}}$/S7	AD0	有效的数据引脚	操作
0	0	AD15~AD0（一个总线周期同时访问奇体和偶体，从奇地址单元读/写数据的高 8 位，从偶地址单元读/写数据的低 8 位）	从偶地址读/写一个字
1	0	AD7~AD0	从偶地址读/写一个字节
0	1	AD15~AD8	从奇地址读/写一个字节
0 1	1 0	AD15~AD8（第一个总线周期从奇地址读写字数据的低 8 位） AD7~AD0（第二个总线周期从偶地址读写字数据的高 8 位）	从奇地址读/写一个字

2) 最小方式下的引脚

8086 最小方式的基本配置如图 2.4所示，系统主要由 8086CPU、时钟发生器 8284、地址锁存器及数据总线收发器组成。

由于地址与数据、状态线分时复用，因此系统中需要地址锁存器将地址锁存。数据线连至内存及外设，负载大，需用数据总线收发器做驱动。而控制总线一般负载较轻不需要驱动，故直接从 8086 引出。

最小方式下第 24~第 31 引脚信号简介如下：

- $\overline{\text{INTA}}$（Interrupt Acknowledge） 中断响应信号输出线。中断响应信号低电平有效，当 CPU 响应外设中断申请时，发出两个连续有效的 $\overline{\text{INTA}}$ 信号。

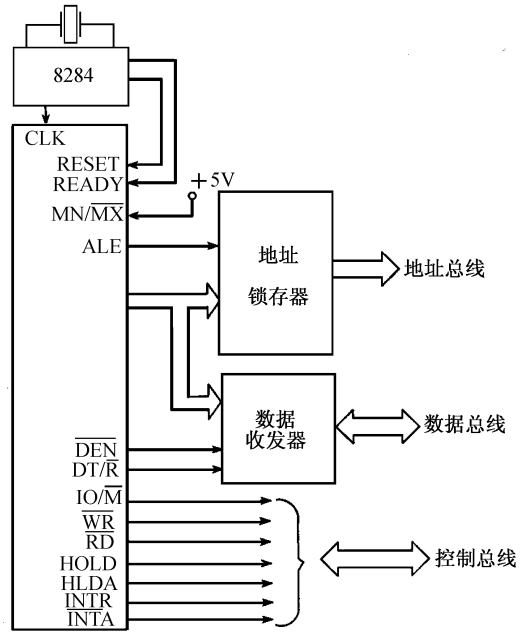

图 2.4 8086 最小方式下的配置

- ALE（Address Latch Enable） 地址锁存允许信号输出线。地址锁存允许信号是 8086 提供给地址锁存器的控制信号，高电平有效。
- $\overline{\text{DEN}}$ （Data Enable） 数据允许信号输出线。CPU 发出的数据允许信号作为数据总线收发器的输出允许信号。当 CPU 处于 DMA 方式时，此线浮空。
- DT/$\overline{\text{R}}$ （Data Transmit/Receive） 数据发送/接收信号输出线。该信号用来控制数据

总线收发器的传送方向。该引脚为高电平，CPU 向内存或 I/O 端口发送数据；该脚为低电平，CPU 从内存或 I/O 端口接收数据；当 CPU 处于 DMA 方式时，DT/$\overline{R}$ 被置为浮空态。

- M/$\overline{IO}$（Memory/Input and Output） 存储器/输入和输出控制输出线。该引脚为高电平时，表示 CPU 正与内存进行数据传输；当其为低电平时，表示 CPU 正与 I/O 端口进行数据传送；当 CPU 处于 DMA 方式时，此线浮空。
- $\overline{WR}$（Write） 写信号输出线。该引脚低电平表明 CPU 正在对内存或 I/O 端口进行写操作。当 CPU 处于 DMA 方式时，该信号置为浮空态。M/$\overline{IO}$，$\overline{WR}$ 和 $\overline{RD}$ 信号的组合、对应的操作及 8086 指令之间的关系如表 2.3所示。

表 2.3 M/$\overline{IO}$，$\overline{WR}$ 和 $\overline{RD}$ 信号的组合和对应的操作

	$\overline{WR}$	$\overline{RD}$	对应的操作	相关的指令举例
0	0	1	写 I/O 端口	OUT 43H，AL
0	1	0	读 I/O 端口	IN AL，DX
1	0	1	写内存	MOV [BX]，AX
1	1	0	读内存	MOV AX，[1234H]
X	0	0	无效信号	-
X	1	1	非存储器或 I/O 读写操作	MOV AX, BX

- HOLD（Hold Request） 总线保持请求信号输入线。当 8086CPU 外的总线主设备要求占用总线时，通过该引脚向 CPU 发一个高电平的总线保持请求信号。
- HLDA（Hold Acknowledge） 总线保持响应信号输出线。当 CPU 接收到 HOLD 信号后，便发出高电平有效的 HLDA 信号给以响应。此时，CPU 让出总线控制权，发出 HOLD 请求的总线主设备获得总线的控制权。

3) 最大方式下的引脚

8086 最大方式的基本配置如图 2.5所示。

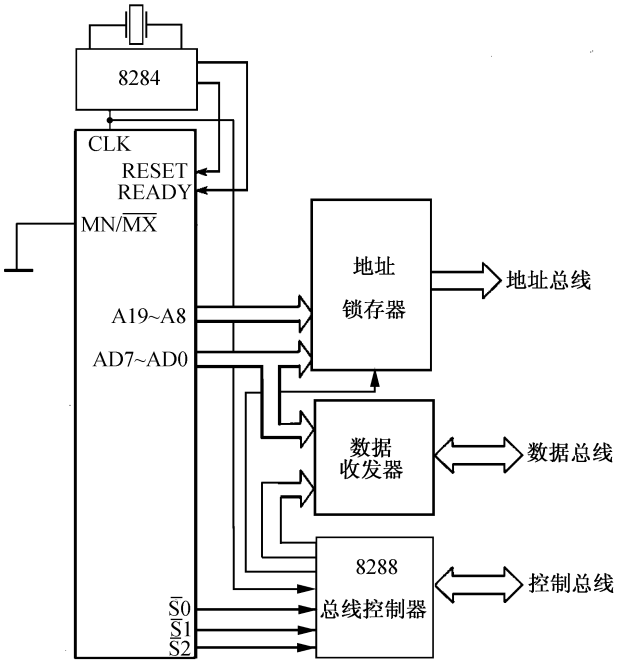

图 2.5 8086 最大方式下的配置

比较最大方式和最小方式的基本配置图，可以看出，最大方式和最小方式有关地址总线

和数据总线的电路部分基本相同，即都需要地址锁存器及数据总线收发器，而控制总线的电路部分有很大的差别。在最小方式下，控制总线直接从 8086 得到，不需外加电路；最大方式是多处理器方式，需要协调主处理器和协处理器的工作。因此，控制总线不能直接从 8086 引脚引出而需外加总线控制器 8288。

下面介绍最大方式下有关引脚的定义。

- QS1 和 QS0（Instruction Queue Status） 指令队列状态输出线。QS1 和 QS0 两个信号电平的不同组合指明了 8086 内部指令队列的状态，当 QS1QS0 为 00, 01, 10 和 11 的时候，分别代表指令队列无操作、从指令队列的第一个字节中取走代码、队列为空和除第一字节外还取走了后续字节中的代码。
- $\overline{S2}$，$\overline{S1}$ 和 $\overline{S0}$（Bus Cycle Status） 总线周期状态信号输出线。低电平有效的三个状态信号连接到 8288 总线控制器的输入线，8288 对这些信号进行译码后产生内存及 I/O 端口的读/写控制信号。三个状态信号的代码组合、对应的操作及 8086 产生的控制信号见表 2.4。

表 2.4 S2，S1 及 S0 的组合与相应操作

$\overline{S2}\ \overline{S1}\ \overline{S0}$	对应的操作	8288 产生的控制信号	相关的指令举例
000	发中断响应信号	$\overline{INTA}$	无
001	读 I/O 端口	$\overline{IROC}$	IN AL，DX
010	写 I/O 端口	$\overline{IOWC}$ 和 $\overline{AIOWC}$	OUT DX，AL
011	暂停	无	NOP
100	取指令	$\overline{MRDC}$	无
101	读内存	$\overline{MRDC}$	MOV AX，[1234H]
110	写内存	$\overline{MWTC}$ 和 $\overline{AMWC}$	MOV [BX]，AX
111	无效	无	无

对于表中的前七种情况，$\overline{S2}\ \overline{S1}\ \overline{S0}$ 三个状态信号中至少有一个为有效的低电平，每一种情况都对应一种总线操作。而第八种情况，$\overline{S2}\ \overline{S1}\ \overline{S0}$ 皆为高电平，此时一个总线过程就要结束，另一个新的总线周期还未开始，通常称这种状态为无效状态。

- $\overline{LOCK}$ 总线封锁信号输出线。当 $\overline{LOCK}$ 信号为低电平时，系统中其他的总线主设备不能获得系统总线的控制权。$\overline{LOCK}$ 信号由前缀指令 LOCK 产生，LOCK 指令后面的一条指令执行完后，$\overline{LOCK}$ 信号失效。另外，在 DMA 期间，$\overline{LOCK}$ 浮空。
- $\overline{RQ}/\overline{GT1}$ 和 $\overline{RQ}/\overline{GT0}$（Request/Grant） 总线请求信号输入/总线请求允许信号输出线。这两个信号端可供 8086 以外的 2 个总线主设备向 8086 发送使用总线的请求信号 $\overline{RQ}$（相当于最小方式时的 HOLD 信号）。而 8086 在现行总线周期结束后让出总线，发出总线请求允许信号 $\overline{GT}$（相当于最小方式时的 HLDA 信号），此时，外部的总线主设备便获得了总线的控制权。$\overline{RQ}/\overline{GT1}$ 比 $\overline{RQ}/\overline{GT0}$ 的优先级高。在 IBM PC 及 PC/XT 中，8086 的 $\overline{RQ}/\overline{GT1}$ 接至 8087 的 $\overline{RQ}/\overline{GT0}$ 端。

3. 8088 与 8086 引脚的不同之处

8086 面世之际，市面上正式 8 位微处理器为主流处理器的时候，因此，当时所有的外部设备群采用的 8 位数据线，为了方便原来的 8 位机用户，1979 年 Intel 公司推出了内部 16 位结构、外部数据总线为 8 位的 Intel 8088，其指令系统和 8086 兼容。IBM 公司利用 Intel 8088 微处理器为核心研制的 IBM PC 成为个人计算机的主流机种之一。

8088 与 8086 内部结构基本相同，外部都采用 40 引脚双列直插式封装。图 2.6所示是 8088 引脚图，括号内为最大方式下引脚的定义。

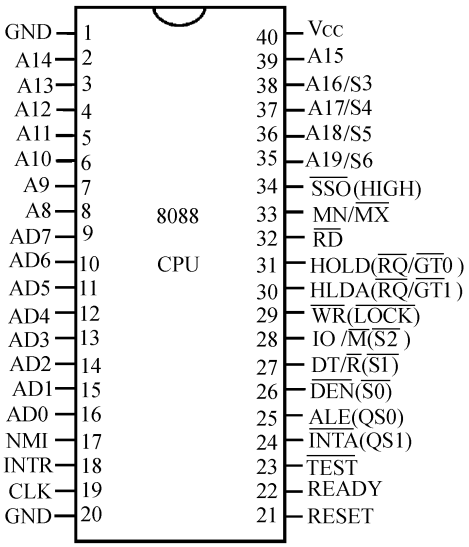

图 2.6 8088 引脚

尽管 8088 与 8086 的大部分引脚是一样的，但依然存在一些不同之处。

- 数据与地址的复用线，8086 为 16 根（AD_{15}~AD_0），而 8088 为 8 根（AD_7~AD_0）。正是由于 8088 对外数据线只有内部数据线的一半，每次只能传输一半的数据，所以，8088 被称为准 16 位结构的微处理器。
- 8088 和 8086 的第 28 脚的功能是相同的，都是输入和输出/存储器控制输出线，但有效电平的高低定义不同，8086 的第 28 脚为 M/$\overline{IO}$，而 8088 的第 28 脚为 IO/$\overline{M}$，电平与 8086 正好相反。
- 8088 和 8086 的第 34 脚有所不同。8086 定义为$\overline{BHE}$/S7，而在最小方式下， 8088 第 34 脚为$\overline{SSO}$（System Status Output）是系统状态信号输出线。在最大方式下，8088 的第 34 脚保持高电平。